

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)**ScienceDirect**

Procedia Computer Science 84 (2016) 57 – 64

**Procedia**  
Computer Science

7th International conference on Intelligent Human Computer Interaction, IHCI 2015

## Digitally Transparent Interface using Eye Tracking

Jai Prakash, Purushottam Swami, Gaurav Khandelwal\*, Mandakinee Singh, Ajay Vijayvargiya

*Samsung R&D Institute India, Bagmane Constellation Business Park, Marathahalli, Bengaluru, Karnataka 560037, India*

---

### Abstract

See-through displays allows user to view the scene behind the display. See-through displays are playing a crucial role in shaping augmented reality applications. However, the cost of the transparent displays are higher than the normal displays. In this paper, we disclose a low cost digitally transparent display using a depth camera, two color cameras and a regular display. The method described in this paper tracks the eyes of the user and finds the corresponding distance from the depth sensor and renders the rear camera preview of the tablet or mobile in such a way that the user perceives it as a transparent display. This method has the potential to change the augmented reality applications in tablets or mobiles from a video see-through experience to a glass see-through experience.

© 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the Organizing Committee of IHCI 2015

**Keywords:** Eye tracking, Depth sensors, Augmented reality, Transparent displays ;

---

### 1. Introduction

Augment reality (AR) aims at adding computer generated virtual objects on a live view of the world to make it more substantial. The AR systems can be mainly classified into direct AR systems and indirect AR systems. Direct AR systems don't obstruct user's line of sight. Head Mounted displays, projector based AR and transparent displays fall under this category. In these type of systems the user witnesses a real world see-through experience. On the other hand, the indirect AR systems obstruct the user's line of sight and render viewpoint from another source like camera. The AR applications in hand held devices like tablets and mobiles fall under this category.

The direct AR systems are sophisticated and specialized devices, meant solely for augmented reality applications. The Hololens, Metaglasses and Google glass are some of the examples of direct AR systems. These devices are a powerful tool for an immersive experience but they need to be worn by the user which causes a hindrance. The projector based AR solves this problem by projecting the virtual object directly onto the surfaces, however, this system is not mobile. Another approach which is frequently applied to make it mobile, is by using pico-projectors<sup>2</sup> and carrying it along. On the other hand, the transparent displays provide an immersive experience in augmented reality. The transparent displays are diminishing in size to be compatible with hand-held devices making it mobile. However, the cost of the transparent displays is far more higher than the normal 2D displays.

---

\* Email: [gaurav.k7@samsung.com](mailto:gaurav.k7@samsung.com), Tel.: +91-962-003-3044, Fax: +91-804-181-9000.

E-mail address: [jprakash@andrew.cmu.edu](mailto:jprakash@andrew.cmu.edu), [swamipuru@gmail.com](mailto:swamipuru@gmail.com), [gaurav.k7@samsung.com](mailto:gaurav.k7@samsung.com), [ajay.v@samsung.com](mailto:ajay.v@samsung.com)

The indirect AR systems do not provide an immersive experience as the direct AR systems but are known for their flexibility and mobility. The augmented reality in hand-held devices like tablets and mobiles provide a video experience by capturing the real world from the camera and then augmenting the virtual object on the camera feed.

### Nomenclature

AR	Augmented reality
2D	Two dimensional
3D	Three dimensional
FOV	Field of view
ROI	Region of interest

Transparent displays provide transparency to view the background behind the device. However, for interaction or to display overlays over the scene, which is being viewed through the transparent display, requires additional sensors which tracks the environment and the user<sup>2, 3</sup>. Tracking the user's eyes helps the system to find the line of sight of the user. The sensors sensing the environment helps to find the subject on which the operation has to be performed.

Eye tracking coupled with 2D displays has been used for 3D visualization in literature. To give an illusion of physical presence of the virtual objects, the user's view point is detected and the rendering of virtual objects is performed accordingly. This dynamic perspective feature<sup>2</sup> is commercially available in Amazon's firephone, which explores 3D visualization in the user interface. In Firephone, the user interface changes depending on the user's eyes position with respect to the phone.

To attain a digitally transparent interface, the dynamic perspective<sup>2</sup> feature can be extended to the rear camera preview. In this paper, we propose a novel method to create a digitally transparent interface using eye-tracking and accordingly cropping the portion of the rear camera view-field that is occluded by the tablet.

The paper is structured as follows. In the next section we describe the idealistic model, the approximate model and the system used for digital transparency. The later section presents the calculations required for the model. Finally, the results are presented along with a discussion on the limitations of approximate model.

## 2. Digital transparency

The digital transparency is obtained when the user is able to view the real world occluded by the display device. In case of the transparent display, the display itself allows viewing the real world behind the screen. Whereas, in case of conventional 2D displays, an opaque object, a camera can be placed behind the 2D display to capture the scene behind and render it on the 2D display to create an impression of digital transparency. We propose an **idealistic model** and **approximate model** for obtaining digital transparency, which are discussed in detail in subsequent sections.

### 2.1. Idealistic model for digital transparency

The idealistic model is the one in which the rear camera is located at the center of the tablet and the background scene occluded by the tablet is detected accurately. This model requires awareness about the user's eye position as well as the background scene using depth sensors for exact fitting of the occluded portion. Fig. 1a illustrates the idealistic model showing the overlapping of the two pyramidal structures. The **eye pyramid** has its vertex as the user's eye and its base consists of the peripheral of the tablet device. The **camera pyramid** comprises of camera as the vertex and horizontal and vertical FOV as the faces of the pyramid extending till infinity (i.e. base is determined by presence of object in viewfield of the camera). The occluded region can be found by extending the eye pyramid and finding the region of intersection onto the base of camera pyramid.

The idealistic model necessitates sensing of the background requiring sophisticated depth sensors and also heavy computational power to deal with the depth data. However, we can approximate the model by using a 2D camera behind the display and depth sensors to get distance of the user's eyes.

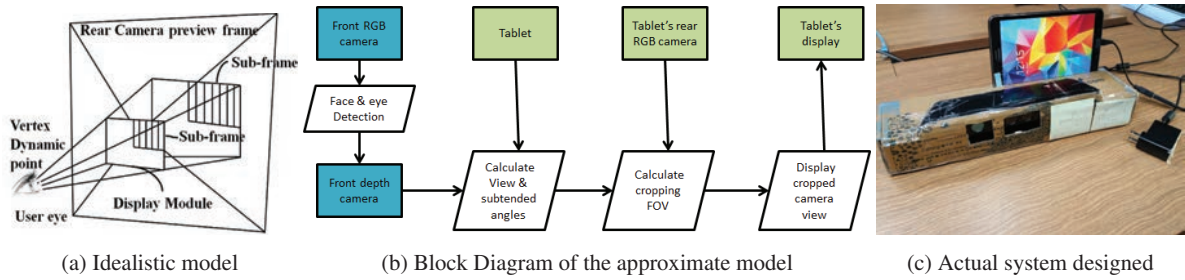


Fig. 1: The system

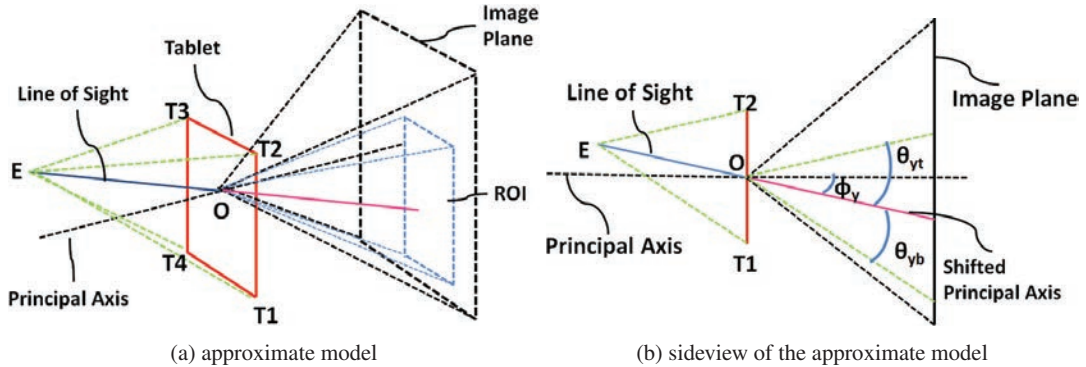


Fig. 2: Models proposed for obtaining digital transparency

## 2.2. Approximate model for digital transparency

The approximate model is not aware of the background scene and hence it does not require any depth sensor to map the scene behind the display. The occluded background can be approximated by cropping the tablet's rear camera view with the same angle as subtended by the user's eyes with the tablet. Moreover, in this model, center( $O$ ) of the tablet [ $T_1, T_2, T_3, T_4$ ] need not be aligned with the camera center as shown in Fig. 2a. This model is apt for practical purpose and can easily be adapted to tablets.

The position of the user's eyes are calculated in three-dimensional space with respect to the tablet's center. The angle subtended by the line-of-sight to the tablet's center is the **view angle** of the user. The principal axis of tablet's rear camera, which is ideally normal to the tablet plane, is shifted by the view angle which is parallel to the user's line-of-sight. The horizontal and vertical angles subtended by the eyes with the sides of the tablet determine the horizontal and vertical FOV cropping of the rear camera preview. The details are described in section 3.

## 2.3. System Overview

The block diagram, shown in Fig. 1b), manifests the hardware devices and the software modules required to realize the system. Fig. 1c shows the actual system used for achieving digital transparency using a tablet and a depth camera. We have used *Samsung galaxy Tab 4 (SM-T331)* having dimensions  $210 \times 124 \times 8mm$ . Since the front 3D camera support is not there in the tablet, *Xbox-360 Kinect sensor* is used to determine user's 3D position (which gives very good results in indoor environments) making it apt for our practical demonstration. The Kinect sensor is powered with an external power supply and connected to the tablet using USB OTG (on-the-go) cable. The tablet is rooted to give read and write permission to the USB device.

The Kinect sensor is fixed in a frame along with the tablet as shown in Fig. 1c. The approximate offsets along the  $x$ ,  $y$  and  $z$ -axis are calculated. Also, there is an angular error while fixing the frame, which is compensated in the software using translation and rotation matrices as calculated in section 3.3. The Linux drivers for the Kinect sensors

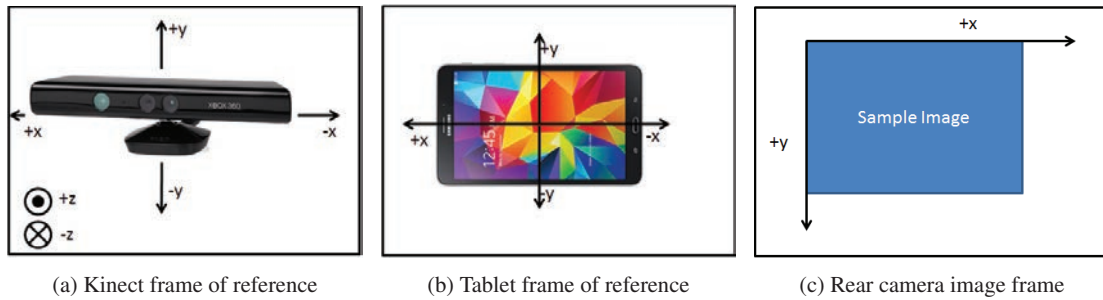


Fig. 3: Frame of Reference

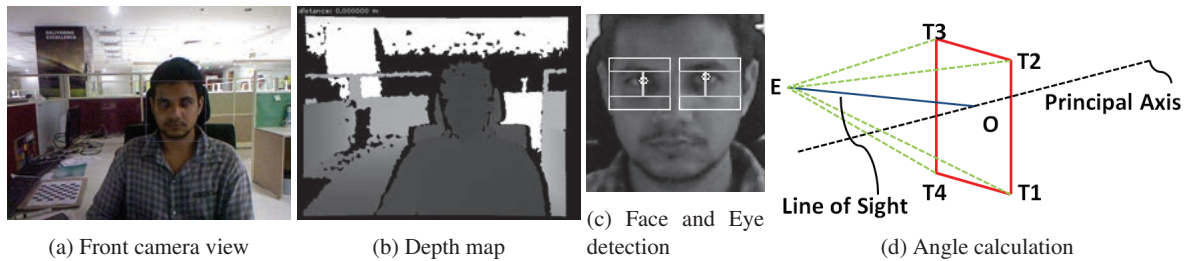


Fig. 4: Eye detection and angle calculation

are ported on Android and an Android application is developed along with OpenCV library to demonstrate the digital transparency.

The front facing Kinect sensor tracks the position of the user's eyes in real-time along with the depth information. The view angle and the angles subtended by the user's eyes with the sides of the tablet are also calculated. These angles are later used to find the horizontal and vertical FOV cropping area for the tablet's rear camera to create the impression of digital transparency. The block diagram in Fig. 1b clearly shows the interactions between different software and hardware modules.

### 3. Design and Implementation

#### 3.1. Frame of reference

For designing the complete system, the data from each of the hardware devices is collected and the correlation between the collected data in the corresponding frames of reference is derived, as shown in Fig. 3. The coordinate system of the Kinect frame of reference is shown in Fig. 3a. Tablet frame of reference is the key to calculate the view angle and the angles subtended by the eyes. The center of the tablet is considered as the origin of the coordinate system as shown in Fig. 3b, whereas rear camera image frame has its origin at the top left corner (Fig. 3c).

The required translation offsets between the Kinect frame of reference and the tablet frame of reference is calculated manually or by using simple calibration techniques. The translation from Kinect frame of reference to tablet frame of reference is done by simple co-ordinate transformation (using translation and rotation). The pin-hole camera model is used to find the correlation between tablet frame of reference and the tablet's rear camera view<sup>7</sup>.

#### 3.2. Eye detection

The essential requirement of the system is tracking user's eye position in real-time. A cascade classifier is used for face detection followed by **Fabian Timm's Algorithm**<sup>7</sup> to track the user's eye-center in real-time. The image captured by the front facing Kinect's RGB camera, as shown in Fig. 4a, is converted into gray scale image and the

user's face is detected. Then the image is passed through a low-pass Gaussian filter, to avoid false outliers. Based on anthropometric relationship<sup>2</sup>, the approximate relative position of the eyes with respect to the user's face are estimated. Following the Fabian Timm's Algorithm, the gradients for both the left and the right eyes are calculated to detect eye's center. The results are shown in Fig. 4c.

To calculate the 3D coordinates of the eyes with respect to the Kinect sensor, the eye coordinates obtained from the RGB camera, are mapped to the depth map (Fig. 4b). Later, the coordinates are translated from the Kinect frame of reference to the tablet frame of reference.

### 3.3. Translation and Rotation matrix

In practical scenario, the Kinect sensor and the tablet cannot be aligned perfectly. Moreover, angular shift may arise while mounting the devices on the frame holder. Hence, translation and rotation are required to shift eye's coordinate from Kinect frame of reference to the tablet frame of reference. Let  $x_{offset}$ ,  $y_{offset}$  and  $z_{offset}$  be the offsets of Kinect's origin with respect to tablet's coordinate system and  $(\alpha, \beta, \gamma)$  be the angular rotation of the Kinect's axis with respect to tablet's axis. In our experiment  $\alpha = \beta = 0$ , so the eye's coordinates can be mapped from Kinect frame of reference to the tablet frame of reference with Eqn 1.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_{offset} \\ 0 & \cos(\gamma) & -\sin(\gamma) & y_{offset} \\ 0 & \sin(\gamma) & \cos(\gamma) & z_{offset} \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (1)$$

### 3.4. View angle and subtended angle calculations

In tablet frame of reference, the center(**O**) of the tablet is the origin as shown in Fig. 4d. **E** represents user's eye position and **T**<sub>1</sub>, **T**<sub>2</sub>, **T**<sub>3</sub>, **T**<sub>4</sub> are the corners of the tablet. The coordinate axes x,y and z are represented as  $\vec{i}$ ,  $\vec{j}$  and  $\vec{k}$  unit vectors respectively. Let  $\vec{ON}$  be the normal vector to the tablet along the z-axis.  $\vec{EO}$  represents the line of sight from the user's eye(**E**) to the tablet center(**O**). To calculate the horizontal and vertical angles subtended by the eye with the tablet we need a reference point from the left, right, top and bottom of the tablet. For simplicity, corner **T**<sub>1</sub> is considered for right-bottom reference and corner **T**<sub>3</sub> is considered for left-top reference. The vectors represented by these two corners are  $\vec{ET}_1$  and  $\vec{ET}_3$ .

To calculate the horizontal and vertical angle subtended by the eye with tablet, the angles are calculated in the  $x-z$  and  $y-z$  planes. The view angle is calculated as the angle between vectors  $\vec{ON}$  and  $\vec{EO}$ . The view angle is calculated as  $\phi_x$  and  $\phi_y$  in  $x-z$  and  $y-z$  planes. Similarly, the eye's angle subtended with the tablet is calculated with vectors  $\vec{ET}_1$  and  $\vec{ET}_3$  with  $\vec{EO}$ . The corresponding projection angles are given by  $\theta_{xl}$ ,  $\theta_{xr}$ ,  $\theta_{yt}$  and  $\theta_{yb}$  for the left, right, top and bottom edge respectively.

- $\phi_x$  : angle between projection of  $\vec{EO}$  and  $\vec{ON}$  on  $x-z$  plane.

$$\phi_x = \cos^{-1} \left( \frac{\vec{EO} \cdot \vec{k}}{\|(\vec{EO} \cdot \vec{i})\vec{i} + (\vec{EO} \cdot \vec{k})\vec{k}\|} \right) \quad (2)$$

- $\theta_{xl}$  : angle between projection of  $\vec{EO}$  and  $\vec{ET}_3$  on  $x-z$  plane defining the top limit of FOV.

$$\theta_{xl} = \cos^{-1} \left( \frac{((\vec{ET}_3 \cdot \vec{i})\vec{i} + (\vec{ET}_3 \cdot \vec{k})\vec{k}) \cdot ((\vec{EO} \cdot \vec{i})\vec{i} + (\vec{EO} \cdot \vec{k})\vec{k})}{\|((\vec{ET}_3 \cdot \vec{i})\vec{i} + (\vec{ET}_3 \cdot \vec{k})\vec{k})\| \times \|((\vec{EO} \cdot \vec{i})\vec{i} + (\vec{EO} \cdot \vec{k})\vec{k})\|} \right) \quad (3)$$

- $\theta_{xr}$  : angle between projection of  $\vec{EO}$  and  $\vec{ET}_1$  on  $x-z$  plane defining the bottom boundary of FOV.

$$\theta_{xr} = \cos^{-1} \left( \frac{((\vec{ET}_1 \cdot \vec{i})\vec{i} + (\vec{ET}_1 \cdot \vec{k})\vec{k}) \cdot ((\vec{EO} \cdot \vec{i})\vec{i} + (\vec{EO} \cdot \vec{k})\vec{k})}{\|((\vec{ET}_1 \cdot \vec{i})\vec{i} + (\vec{ET}_1 \cdot \vec{k})\vec{k})\| \times \|((\vec{EO} \cdot \vec{i})\vec{i} + (\vec{EO} \cdot \vec{k})\vec{k})\|} \right) \quad (4)$$

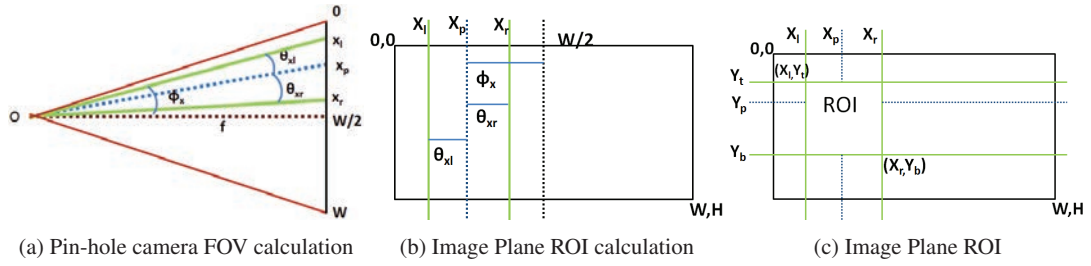


Fig. 5: FOV and ROI calculation

- $\phi_y$  : angle between projection of  $\vec{EO}$  and  $\vec{ON}$  on  $y - z$  plane.

$$\phi_y = \cos^{-1} \left( \frac{\vec{EO} \cdot \vec{k}}{\|(\vec{EO} \cdot \vec{j})\vec{j} + (\vec{EO} \cdot \vec{k})\vec{k}\|} \right) \quad (5)$$

- $\theta_{yl}$  : angle between projection of  $\vec{EO}$  and  $\vec{ET}_3$  on  $y - z$  plane defining the left limit of FOV.

$$\theta_{yl} = \cos^{-1} \left( \frac{((\vec{ET}_3 \cdot \vec{j})\vec{j} + (\vec{ET}_3 \cdot \vec{k})\vec{k}) \cdot ((\vec{EO} \cdot \vec{j})\vec{j} + (\vec{EO} \cdot \vec{k})\vec{k})}{\|((\vec{ET}_3 \cdot \vec{j})\vec{j} + (\vec{ET}_3 \cdot \vec{k})\vec{k})\| \times \|((\vec{EO} \cdot \vec{j})\vec{j} + (\vec{EO} \cdot \vec{k})\vec{k})\|} \right) \quad (6)$$

- $\theta_{yb}$  : angle between projection of  $\vec{EO}$  and  $\vec{ET}_1$  on  $y - z$  plane defining the right limit of FOV.

$$\theta_{yb} = \cos^{-1} \left( \frac{((\vec{ET}_1 \cdot \vec{j})\vec{j} + (\vec{ET}_1 \cdot \vec{k})\vec{k}) \cdot ((\vec{EO} \cdot \vec{j})\vec{j} + (\vec{EO} \cdot \vec{k})\vec{k})}{\|((\vec{ET}_1 \cdot \vec{j})\vec{j} + (\vec{ET}_1 \cdot \vec{k})\vec{k})\| \times \|((\vec{EO} \cdot \vec{j})\vec{j} + (\vec{EO} \cdot \vec{k})\vec{k})\|} \right) \quad (7)$$

Eqn(s). 2-7 give the shift angle in the principal axis and the angles to which the tablet's rear camera view field has to be cropped. The angles obtained in tablet frame of reference need to be correlated to tablet's rear camera image frame of reference using the pin-hole camera model<sup>2</sup>.

### 3.5. Rear camera FOV and ROI calculations

Fig. 5a shows the pin-hole camera model with focal length  $f$ . The intrinsic parameters like  $f$  for a given image size (width  $W$  and height  $H$ ) of the camera can be calculated using the chess-board calibration technique<sup>2</sup>. The principal axis is shifted by angle  $\phi_x$  which maps to  $X_p$  in the  $x$ -coordinate of image frame. Similarly, the left and right cropping FOV angles ( $\theta_{xl}$  and  $\theta_{xr}$ ) maps to  $X_l$  and  $X_r$  as labelled in Fig. 5b. Same calculations apply for the vertical directions resulting in  $Y_p$ ,  $Y_t$  and  $Y_b$ .  $X_l$ ,  $X_r$ ,  $Y_t$  and  $Y_b$  define the ROI of the tablet's rear camera for digital transparency.

- $X_l$  : Left-most  $x$ -coordinate of the image to be cropped

$$X_l = \frac{W}{2} - f \times \tan(\phi_x + \theta_{xl}) \quad (8)$$

- $X_r$  : Right-most  $x$ -coordinate of the image to be cropped

$$X_r = \frac{W}{2} - f \times \tan(\phi_x - \theta_{xr}) \quad (9)$$

- $Y_t$  : Top-most  $y$ -coordinate of the image to be cropped

$$Y_t = \frac{H}{2} - f \times \tan(\phi_y + \theta_{yt}) \quad (10)$$



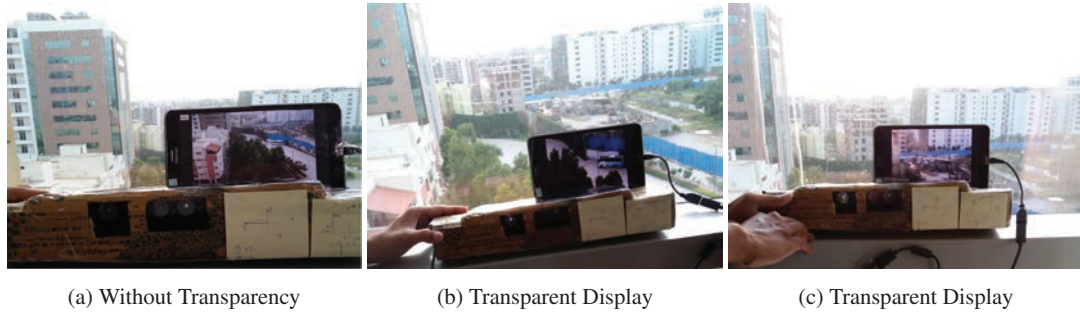


Fig. 6: Results Obtained

- $Y_b$  : Bottom-most y-coordinate of the image to be cropped

$$Y_b = \frac{H}{2} - f \times \tan(\phi_y - \theta_{yb}) \quad (11)$$

The ROI region (Fig. 5c) is rectangle area starts with  $(X_l, Y_l)$  and width is  $X_r - X_l$ , height is  $Y_t - Y_b$ . This ROI is now displayed on the full-screen of the tablet for obtaining digital transparency.

### 3.6. Practical results

As explained above, eye-tracking and angle computation works in real-time whereas rendering is constrained due to OpenCV, delivering  $\approx 10$ fps. The practical results obtained are shown in Fig. 6. Fig. 6a shows tablet's rear camera complete-preview without transparency and, Fig. 6b, Fig. 6c demonstrate the digital transparency phenomenon for an outdoor environment. The results obtained are very close to the digital transparency as expected and exhibits potential equivalence to the transparent displays. We are therefore able to mimic the see-through displays using a low cost system.

## 4. Conclusion and Future work

This paper presents low cost alternative for transparent displays using normal 2D display, an assembly of depth camera and rear facing RGB camera. It also presents true see-through techniques, that can be used for AR-HMD or AR hand held devices. It enumerates the required hardware components and their alignment. Aforementioned assembly comprises of a tablet with rear camera sensor and a kinect sensor as front facing depth camera for tracking user's eyes. Kinect sensor is connected to the tablet using OTG(On-the-go) cable. An open-source algorithm is used, which detects human face and then approximates the eye-position to detect eye-center. This eye-position is mapped with the depth map obtained from kinect sensor, to calculate 3D position of the eye with respect to the tablet's center. The user's line of sight and the angles subtended on the tablet's boundaries are calculated. These angles help to determine the ROI region. ROI region is the area which is occluded by the tablet device from the user's line of sight. Finally, the occluded area is displayed over the full-screen that makes the tablet device as virtual transparent display and creates dynamic perspective view.

Using the proposed techniques and with embedded depth sensors (which can easily be embedded in the tablet), a hand held device like tablet or mobile can be used for sophisticated tasks like augmented reality without requiring any specialized device. Graphical object can be added on the virtually transparent interface for more immersive experience in augmented reality.

To provide an immersive and enhanced user-experience, following things can be considered for future implementation:

- **Single display device** - Apart from the above mentioned errors, the system was limited to only one user and couldnot be modelled for multiple users. However, in AR-HMD devices or AR-glasses, this problem can be solved by rendering the display according to individual users.
- **Depth perception** - This limitation is introduced since there is only one camera at the rear and hence the depth perception cannot be created. Using a stereo display and stereo capture system, depth perception can be created to align the digital display with real objects. Thus, graphical objects can be augmented with accurate depth perception to improve user-experience.
- **User registration** - The current system shows unpredictable behavior in case if multiple faces gets detected simultaneously. In this scenario, it becomes indeterministic that which face is under consideration. Using the face registration process one can avoid such behaviors. Face registration will make sure that once a face is detected and registered, the system will not consider any other face till the registered face goes out of context.

However, we experienced some practical limitation during the implementation of digital transparency which are summarized in the following section.

#### 4.1. System Constraints

The approximate model contains numerous kinds of errors due to various assumptions, approximations and physical differences with the idealistic model. The errors are broadly classified as follows:

- **Camera positioning error** - The idealistic model assumes the rear camera to be present at the center of the tablet, but in practical the camera is present in the upper half of the tablet which introduces some parallax error while viewing the background.
- **Parallax error due to eye** - Since the model assumes only one eye vertex, the results are good when viewed from one eye. However, in practical case of two eyes, the mid-point of two eyes is taken as the vertex of the pyramid. This assumption casts parallax error when the background is viewed from either eye.
- **Perspective error** - The scene captured from eye's point of view is different from the scene captured from camera's point of view resulting in a perspective change which looks unrealistic. The perspective change becomes insignificant for farther objects because the distance between user and camera is very less as compared to the distance between the camera and scene.
- **FOV overshooting** - This phenomenon occurs when the user views the scene that is outside of camera view-field range, or the user's viewing angles reach outside of the tablet's rear camera FOV extremities. This limitation can be overcome by using a wide angle rear camera.

## References

1. Pranav Mistry and Pattie Maes. Sixthsense: A wearable gestural interface, 2009.
2. I. YUN, J. Seo, and C.S. Lee. Transparent display apparatus and display method thereof, February 6 2014. US Patent App. 13/957,027.
3. Jinha Lee, Alex Olwal, Hiroshi Ishii, and Cati Boulanger. Spacetop: Integrating 2d and spatial 3d interactions in a see-through desktop environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 189–192, New York, NY, USA, 2013. ACM.
4. H. Cheng and C. Newman. Eye tracking enabling 3d viewing on conventional 2d display, April 22 2014. US Patent 8,704,879.
5. S.A. Mann, J. Bertolami, M.L. Bronder, M.A. Dougherty, R.M. Craig, and J.A. Tardif. Dynamic perspective video window, February 19 2013. US Patent 8,379,057.
6. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
7. Fabian Timm and Erhardt Barth. Accurate eye centre localisation by means of gradients. In Leonid Mestetskiy and Jos Braz, editors, *VISAPP*, pages 125–130. SciTePress, 2011.
8. AbuSayeedMd. Sohail and Prabir Bhattacharya. Detection of facial feature points using anthropometric face model. In Ernesto Damiani, Kokou Ytongnon, Peter Schelkens, Albert Dipanda, Louis Legrand, and Richard Chbeir, editors, *Signal Processing for Image Enhancement and Multimedia Processing*, volume 31 of *Multimedia Systems and Applications Series*, pages 189–200. Springer US, 2008.
9. Peter F. Sturm and Stephen J. Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. In *CVPR*, pages 1432–1437. IEEE Computer Society, 1999.